Integrated Decision Support System
For Optimizing The Training And Transition Of Airline Pilots

Field Of The Invention

The invention relates generally to decision support systems, and more particularly to an interactive and integrated decision support system for providing alternative optimized training plans to strategic planners.

Background Of The Invention

Numerous systems have been developed to overcome the complexity of determining when training and transition is to occur, and for which crew members, at which locations, at what times, and with an appropriate allocation of training resources including equipment and instructors.

The prior art systems have included both manual and automated systems with response times ranging from days, to weeks, and even months. Further, such systems have tended to implement a decision making process for providing a single solution, rather than a dynamic, adaptive, decision support system providing alternative solutions for evaluation by a strategic planner. In addition, such prior systems have been represented by models which either are too complex for commercial software solution, or have simplifying assumptions that make them too unrealistic for practical use. Prior systems also have generally been too costly in employee and equipment resources. See "Decision Support Systems-An application in strategic manpower planning of airline pilots" by Peter J. Verbeek, European Journal of Operational Research 55 (1991), pages 368-381, Elsevier Science Publishers B.V. While the Verbeek article does not disclose a decision support system, it does describe the enormous complexities which must be addressed in designing such a system to accommodate the large numbers of constraints and variables that are required for a solution to be realistic. A reference is made in the article to mixed integer models solved by linear programming (LP) which were developed by United Airlines and American Airlines, but fell short of being both realistic and optimal. Verbeek also referred to his own mixed integer model which was admittedly to complex for solution with commercial software, and thus too costly in time.

The article, "Course Planning at Lufthansa Technical Training: Constructing More Profitable Schedules", by Knut Haase, Jorg Latteier, and Andreas Schirmer, INTERFACES 29:5 September-October 1999 (pp. 95-109) likewise does not disclose a specific model, but rather describes the enormity of the problems faced in attempting to design and develop a decision support system which can accommodate continuing data input changes while realizing reduced costs in creating training plans. No indication of creating and placing comparative values on alternative plans to generate multiple training plan options is indicated. Further, no indication of turnaround time is provided.

From the above it may be discerned that the problem of pilot staffing and training is one of the most complex and costly problems facing the major airlines. If not managed effectively, an airline cannot survive, not to mention profit, in the competitive air transportation market.

By way of example, Continental airlines provides both domestic and international service to more than 100 destinations around the world. They operate 325 aircraft of nine different fleet types to fly 1400 daily flights. Their 5000 pilots are stationed at three domestic and two international crew bases. At least twice a year Continental conducts a system bid award. These awards provide an opportunity for pilots to use their seniority to increase their pay and improve their work schedules by changing their position (base, fleet, and status), and a way for the airline to adjust staffing levels in response to retirements, attrition, and changes in their business plan. In an average system bid award, 15-20% of the airline's pilots receive new positions. The problem of taking the 15-20% of Continental's pilots who have received new positions, and finding a training class for each pilot requiring training, an advancement date for each pilot changing position without training, and a release date for each pilot leaving the airline is a very large NP-hard problem which must be solved. Additional complexity for Continental comes from the facts that: pilot positions are interrelated; the timing and number of training classes is variable; minimal length student training schedules must be generated using limited resources; and numerous complicating regulations and business rules related to each pilot's seniority, flight history, and current and future position must be considered.

Continental manpower planners with expert knowledge took more than two weeks to manually generate a single, partial, sub-optimal training plan for ensuring adequate staffing levels with no detailed consideration of costs.

In contrast to prior systems and methods, the invention described and claimed herein is a realistic representation of the real world problem as evidenced by its implementation by Continental Airlines, and is modeled so efficiently that it can be solved in under an hour. An hour is a huge improvement over the time required by the prior art systems, and is a very reasonable amount of time for a planning problem as complex as the one addressed by this invention.

The invention as described herein has been developed by CALEB Technologies Corp. of Austin, Texas, as part of an integrated decision support system, referred to as the ManpowerSolver system, to face this challenge. The ManpowerSolver system manages large volumes of data and employs state-of-the-art optimization modeling and solution techniques to efficiently allocate human and training resources, and attain optimal operational and cost effective performance.

In response to a system bid award, the ManpowerSolver system in accordance with the invention builds a training plan that establishes the timing and number of pilot new hires, training assignments, advancements, and releases. It also determines the number of pilots whose training or release should be postponed, and the flow of pilots across different positions in a manner that ensures adequate staffing levels, minimum cost, and efficient utilization of training resources.

By exercising different parameter settings, multiple high-quality solutions are generated that can be carefully examined before a suitable one is chosen by the user. The substantial time and process savings, however, are dwarfed by the savings derived from implementing an optimized solution that not only provides complete coverage of flights, but also significant dollar savings from reduced staffing costs, pay protection costs, and training costs. Staffing cost savings come from decreases in the number of new pilots hired, postponement of new pilot hires, quicker release of pilots without an award, and providing just-in-time training to minimize pilots being trained long before they are needed. Training cost savings arise from better utilization of expensive resources, reduced time for pilots to complete training, and a reduction in the number of training cycles required. Overall, Continental has estimated a dollar savings in excess of $10 million annually.

The primary objective of the training plans produced by the invention described and claimed herein is to have pilots in place at each combination of base (assigned geography), fleet, and status (seat) to cover all of the published flights. Secondary objectives include minimizing staffing, pay

protection, and training and hiring costs. Constraints that are considered include pilot vacations and absences, available training resources, new hires, seniority, and training policies.

To solve the pilot training and transitioning (staffing) problem the ManpowerSolver system relies on a loosely coupled solution methodology that employs a unique combination of modeling and algorithmic approaches. Innovative aspects of this methodology include a decomposition of the problem into a series of linear and mixed integer programs, and a specialized branch and bound algorithm with custom branch ordering and node bounding techniques. Alternative cost effective training plans are produced and evaluated for selection of the training plan deemed most suitable to the airline.

The system can also be used in a what-if mode to gauge the impact that various operating decisions will have on pilot staffing and training. This tool allows crew planners to be proactive as they face changes in the airline such as new market opportunities, the acquisition and retirement of aircraft and training resources, opening and closing sub-bases, and modification to the number of hours to be flown from different pilot positions to allow the airline to take advantage of business opportunities.

This integrated approach to workforce management that takes into account the broad spectrum of hiring, staffing, training, and absence is the first in the airline industry. The concept and framework as well as some of the solution techniques are general enough to be applied to industries other than airlines that desire a skilled workforce.

Summary Of The Invention

An adaptive, dynamic, automated optimizer system and method for rapid generation of multiple alternative training plans for all pilots of an airline who have received a system bid award, and who have been awarded a change of position, whether by advancement, furlough, or no award, wherein such training plans are derived from an optimized and realistic solution of a mixed integer programming representation of the pilot training/transition problem.

In one aspect of the invention, pilot data and other information received from a user are sorted into segregated lists from which variables and constraints are created, and an MIP Model of the training/transition problem is developed from the variables and constraints.

In another aspect of the invention, an LP relaxation of the MIP Model is first made and solved before solving the MIP Model to determine a first level of feasibility, and the MIP Model is thereafter solved to provide a feasible and optimized solution of the MIP Model in a short period of time.

In yet another aspect of the invention, costs associated with a solution of the LP relaxation of the MIP Model are calculated and used to modify cost factors in the objective function of the MIP Model, thereby providing weightings of both block hour and dollar costs.

In still another aspect of the invention, multiple alternative solutions of the MIP Model may be generated by altering parameters including one or more of sub-base openings, sub-base closings, vacation cancellations, level of importance, costs, capacity, excesses, shortages, and training resource availability.

In a further aspect of the invention, the pilot training/transition problem is represented by a MIP Model comprised of an objective function, with variables and constraints, which is simplistic enough for commercial software solution.

In yet a further aspect of the invention, comparative values are placed on alternative multiple training plans to assist the user in selecting a training plan to be used.

In a still further aspect of the invention, continuing data input changes are accommodated while realizing reduced staffing, pay protection, and training costs in generating multiple, alternative training plans which cover all published flights, and which provide timing and number of pilot new hires, training assignments, advancements, and releases.

In an additional aspect of the invention, the invention may be used in a what-if mode to guage the impact that various operating decisions will have on pilot staffing and training.

In yet an additional aspect of the invention, the solution of the MIP Model is optimized by minimizing the objective function of the MIP Model to have a lowest value within a region defined by the constraints of the MIP Model.

Brief Description Of The Drawings

Additional objects, features and advantages of the present invention will become apparent from the following detailed description when read in conjunction with the accompanying drawings in which:

Figure 1 is a functional block diagram of a training plan optimization system in accordance with the invention;

Figure 2 is a logic flow diagram of the optimizer engine 8 of Figure 1;

Figure 3 is a logic flow diagram of the preprocessing stage 22 of Figure 2;

Figure 4 is a logic flow diagram of the model creation stage 23 of Figure 2; and

Figure 5 is a logic flow diagram of the solution and postprocessing stage 24 of Figure 2.

Description Of Preferred Embodiments

Preferred embodiments of the invention will now be described with reference to the accompanying drawings. In the description, the following terms, whether capitalized or in lower case, shall have the attendant meanings indicated below when used in this specification:

Table I

Term	Definition
System Bid	A seniority-based award in which an airline adjusts staffing levels in response to retirements, attrition, and changes in their business plan by awarding pilots new positions.

System Bid Information	Complete information on a pilot's awarded position from a specific system bid.
Bid Effective Date	Date by which all pilots should be advanced to their awarded positions for a specific system bid.
Average Pay Hours	Average number of hours per month for which a pilot in a certain position is paid.
Utilization	Average number of hours per month during which a pilot works.
Bid Period	A time period roughly equivalent to a calendar month.
Training Capacity	Maximum number of students that can start training for a given fleet in a given bid period.
Pay Protection	Penalty paid to a pilot in certain cases where a junior pilot is advanced ahead of a senior pilot.
Training Plan	Establishes the timing and number of pilot hires, training assignments, advancements, and releases.
Training Schedule	Detailed daily schedule for all students and training resources based on the classes found in the training plan.
Total Shortage	The shortage of pilots measured as the deficit in ability to cover block hours in each pilot position each bid period.
Percentage Shortage	The shortage of pilots measured as a percentage of the total number of pilots needed.
Total Excess	The excess of pilots measured as the overage in ability to cover block hours in each pilot position each bid period.
Percentage Excess	The excess of pilots measured as a perce`ntage of the total number of pilots needed.
MIP Model	The mixed integer programming mathematical model that represents the training and transition problem for airline pilots, and which is used to obtain solutions to this problem.
Key	The unique identifier of a variable or constraint in the model.
Level of Importance	The weighting (1-5) given to each cost factor demonstrating its

	importance relative to other cost factors
Cost Factor	An item used to weigh the value of one solution vs. other solutions
Optimizer Engine	An optimizer or optimizer engine is a program implementing operations research algorithms, and producing an optimized solution based upon user input and system data.
Optimizer Session	A request to an optimizer engine to perform its function.
Optimizer Container	A generic hosting environment in which Optimizer Engines execute Optimizer Sessions.
Objective Function	An objective or objective function is an algebraic expression that combines variables with variable costs.
Status	The seat a pilot is assigned to in an aircraft, for example captain, first officer, or second officer.
Equipment	A type of aircraft.
Pay Protection	Extra pay received by a pilot when certain contractual obligations are not met by the airline.
Displaced Pilot	Pilot who receives a system bid award for a lower paying position.
Straight Displacement Pilot	Pilot who receives a system bid award for a lower paying position that does not require training, only advancement.
Advanced Pilot	Pilot who has moved from an old assignment to a new awarded assignment.
Straight Advance Pilot	Pilot who receives a system bid award that does not require training, only advancement.
Furloughed Pilot	Pilot who receives a system bid award indicating that he will be released from the airline.
No Award Pilot	Pilot having a greater than 60 age who receives a bid award indicating that he must be released from the airline.
Included Pilot	Pilot whose new system bid award involves movement to a new position in the airline, or who is released or furloughed from the airline..
Age 58 Pilot	Pilot who will be 58 or older by the effective date of the system bid.

(58Y)	These pilots have special rules for bidding and pay protection.
Base	A geographical location where pilots are stationed.
Sub-base	A combination of a type of aircraft and a base.
Position	A combination of a sub-base and status, for example a pilot may be a captain flying DC10 aircraft out of the Houston base.
Sub-base Setting	Option for the user to generate a training plan using block hours by fleet or by sub-base.
Block Hours	The scheduled flight hours the airline plans to operate.

The environment in which the invention operates is illustrated in Figure 1, in which user requests and input data are provided by a client 1 to an optimizer system 2 by way of a user interface including an optimizer protocol layer 3. The optimizer protocol layer 3 uses an XML-based transaction protocol via HTTP over TCP/IP to communicate with the optimizer system 2, and to provide input data to a data base 4. In the preferred embodiment, the optimizer system 2 is deployed on a Compaq ProLiant Server with 4 Pentium III 550 MHz processors.

The data base 4 has stored therein information including but not limited to pilot data including system bid award information, average pay hours, pilot utilization, pay protection, pilot vacation and absence information, and retirement criteria; and training information including training capacity by fleet, and operational and contractual constraints affecting training. The data base also stores customer optimization requests and information associated with each request including but not limited to optimization options such as allowing cancellation of vacations, exceeding class capacity, grouping pilots in classes, sub-base settings to solving the training plan problem by fleet or sub-base, and request status information. In the preferred embodiment, the data base 4 is a Microsoft SQL Server 2000 relational database.

The optimizer system 2 is comprised of an optimizer controller 5 which receives user requests from the optimizer protocol layer 3, and notification and optimizer update messages from an optimizer container 6 whose primary function is to communicate with an optimizer data provider 7

and an optimizer engine 8. In the preferred embodiment, the data base 4, the optimizer controller 5, and the optimizer container 6 are part of a computer system operating under the Microsoft Windows 2000 Advanced Server Operating System.

In response to a user request, the optimizer controller 5 creates an optimizer session for storage into the data base 4, and sends a request acknowledgement and an optimizer session reference to the optimizer protocol layer 3 for relay to the user. Thereafter, the optimizer controller 5 notifies the optimizer container 6 of the request for services by way of an RPC call (processing instructions provided by the XML transaction protocol) routed via HTTP over TCP/IP (a transport protocol implementation).

The optimizer container 6 thereupon verifies to the optimizer controller 5 that it has the necessary resources and request parameters to accommodate the user request. The optimizer container 6 then initializes the optimizer data provider 7 to obtain necessary data from the data base 4 for use by the optimizer engine 8. In the preferred embodiment, the data provider 7 and the optimizer engine 8 are deployed as Windows dynamic link libraries.

The optimizer engine 8, which in the preferred embodiment is an optimizer engine solving a training plan problem, implements operations research algorithms to produce an optimized solution based on the received request. In response to the above verification from the optimizer container 6, the optimizer controller 5 updates the optimizer session status stored in the data base 4 to indicate that the user request is being executed. The optimizer communications layer 3, upon polling the optimizer controller 5, is thereby notified of the status of the user request and so informs the client 1. Once the optimizer data provider 7 is initialized, the optimizer container 6 sends the user request to the optimizer engine 8 for execution.

The results obtained from the optimizer engine 8 are received by the optimizer container 6, which forwards the results to the optimizer controller 5. The optimizer controller 5 thereupon updates the optimizer session status to indicate that the user request has been executed, and that the results are available for access by the user.

Upon the user querying the optimizer controller 5 by way of the optimizer protocol layer 3, the user is informed that the user request has been executed and is provided the results generated by the optimizer engine 8.

The invention described and claimed below resides in the optimizer engine 8 in the form of a software program for developing, solving, and interpreting the results of a mixed integer programming (MIP) model that provides an optimized solution to the problem of training and transitioning airline pilots following a system bid award.

A functional block diagram of the optimizer engine 8 is provided in Figure 2, where the logic flow process in accordance with the invention begins at logic step 20, and then proceeds to logic step 21 where data provided by the optimizer data provider 7 of Figure 1 by way of optimizer container 6 is received by an input data module. Such data includes bid information, average pay hours, utilization, pay protection, training capacity, operational and contractual constraints affecting training, variables such as vacation and retirement criteria, and optimizer options such as allowing cancellation of vacations, exceeding class capacity, grouping pilots in classes, and solving the training plan problem by fleet or base.

The logic flow process continues from logic step 21 to logic step 22 where the content of the input data module of logic step 21 is applied in the form of an XML document to a preprocessing stage. The preprocessing stage in turn creates data structures that provide separate lists of pilots which have been sorted in accordance with received user options and other input data. The data structures of logic step 22 then are applied by the logic flow process to a model creation stage at logic step 23, and are operated upon to create a mathematical mixed integer programming model (MIP Model) that represents the pilot training and transition problem mathematically through an objective function and a combination of constraints and variables. Once the MIP Model is created, the logic flow process continues to the model solution and postprocessing stage of logic step 24, where optimal variable values are found for the MIP Model. The optimal variable values then are postprocessed to determine and display the solution results. The solution is referred to as a training plan that is then stored and may be viewed by the user.

The MIP model is presented below followed by detailed logic flow descriptions of each of the stages illustrated in Figure 2.

Model Indices

The indices which are used in the MIP Model variables and equations that follow are defined in Table II.

Table II

Index	Index Description
i	Pilots. For each pilot the type of class he needs to attend is known.
f	Fleets
h	Combinations of Fleet and Status. If the problem is solved by position, then h will correspond to the combination's base, equipment, and status.
t	Bid periods (including an extra bid period for those pilots that will not be trained during the planning horizon due to capacity constraints).
j	A duplicate index used to represent pilots.

Model Sets

The sets which are used in the MIP Model equations that follow are presented in Table III.

Table III

Set	Set Description
$\phi(i)$	Bid periods in which pilot i can complete his training. This set is determined during the preprocessing of variables described above.
$\varphi(f,t)$	Pilots that can start training fleet f in the bid period t.
λ	Set of pilots that need to be advanced.
λ_1	Set of pilots that need to be advanced, not including the 58 year old pilots that hold a bid and could have held a better bid.
λ_2	Set of pilots 58 years old that hold a bid and could have held a better bid.

APP	Set of pilots already pay protected.
NA	Pilots with no-award who will turn age 60 between now and the effective date of the system bid award.
F	Furlough no award pilots.
H(h)	Pilots whose initial position is h (these are pilots that belong to λ,NA,F).
H'(h)	Pilots whose future position is h (these are pilots that belong to λ,NH).
SA(h)	Pilots who are straight advances whose future position is ***h.***
SD(h)	Pilots who are straight displacements whose future position is ***h.***
PP(i)	Pilots who may cause pilot i to be in pay protection. This set is determined by a preprocessing and clustering of pilots with the same bid effective date and future position, and ranking them in seniority order.
58Y	Pilots 58 years old who did not hold a bid but were senior enough to hold a bid, and will be pay-protected.
$PP_{58}(i)$	Pilots who may cause pilot i $\in$58Y to be in pay protection.
L(i)	Length of training for pilot i {PS=2, RQ=1, Upgrade=1,Advance=0}.

Model Parameters

The parameters which are used in the MIP Model equations that follow are defined below.

Table IV

Parameter	Description Of Parameter
CAP_{ft}	Maximum number of students that can start training for fleet f (course training type) in bid period t.
$Initial_{ht}$	Initial number of hours for position h in the bid period t. This value is obtained by multiplying the number of pilots in a given position and bid period by the respective utilization (headcount should consider retirements and absences).
a_i	Pay protection cost paid to pilot i per month, if any (difference between

	his future pay rate and his current pay rate).
uti_{ht}	Utilization per position h per period t.
PBH	Cost associated with block hours.
PS	Level of importance of shortages in block hours in the solution.
PE	Level of importance of excess in block hours in the solution.
Ppay	Level of importance of pay protection cost in the solution.
PNH	Level of importance of new hires cost in the solution.
PNA	Level of importance of no-awards cost in the solution.
PF	Level of importance of furloughs cost in the solution
α_{ht}	Maximum percentage in shortages per position h per bid period t, after which shortages will be minimized.
β_{ht}	Maximum percentage in excess per position h per bid period t, after which excess will be minimized.
MNH_{ht}	Maximum allowable new hires for position ***h*** in bid period ***t***
$NHCost_{ht}$	Cost per new hire pilot advanced to position h in bid period t (computed as number of months between t and the end of planning horizon, times the pay rate , times the average pay hours).
$NACost_{it}$	Cost if pilot $i \in NA$ is released in bid period t (computed as number of months between t and the beginning of planning horizon, times the pay rate, times the average pay hours).
$FCost_{it}$	Cost if pilot $i \in F$ is released in bid period t (computed as number of month between t and the beginning of planning horizon, times the pay rate, times the average pay hours).
$BidEff_i$	Bid effective date for pilot i.
$Blockhrs_{ht}$	Business plan block hours for position h in bid period t.
N	Length of the planning horizon.
G_i	Month in which pilots will start being pay protected due to a more junior pilot in training at the beginning of the planning horizon. If the pilot is

	already pay protected, then $G_i = 0$.
Last	Last month in which the shortages and excess in block hours will be penalized.

Model Variables

The variables appearing in the MIP Model equations that follow are described below.

For Pilots included in the training set ($i \in \lambda$):

(1) $$y_{it} = \begin{cases} 1 & \text{if pilot i is advanced bid period t} \\ 0 & \text{otherwise} \end{cases}$$

The y_{it} variable will only exist for bid periods during which pilot *i* is eligible to complete his training. Some bid periods may not be possible due to vacation or other absences.

For pilots included in the No-Award set ($i \in NA$):

(2) $$y_{NAit} = \begin{cases} 1 & \text{if pilot i is released in bid period t} \\ 0 & \text{otherwise} \end{cases}$$

For pilots included in the Furlough set ($i \in F$):

(3) $$y_{Fit} = \begin{cases} 1 & \text{if pilot i is released in bid period t} \\ 0 & \text{otherwise} \end{cases}$$

RF_i is computed as:

(4) $$RF_i \geq (N - R_{58i} - \sum_{t \in \phi(i)} t y_{it} + R_i),$$

where $N - R_{58i}$ is the month a pilot starts being pay-protected for the position he could have held but did not (for all pilots 58 years old with bid and potential pay protection different from zero); and $\sum_{t \in \phi(i)} ty_{it} - R_i$ is the month the pilot starts being pay-protected for the position he holds. When the position the pilot could have held is better than the one he holds, he can always get at least the R_{58i} and could potentially receive more months of pay protection (i.e. M_i).

Other variables are defined in Table V below.

Table V

Variable	Variable Definition
Y_{NHht}	For pilots included in the new hires set ($i \in NH$), the number of new hires advanced in month t for position h.
z_{ft}	Number of pilots exceeding the training resource capacity in the bid period t for fleet f.
S_{ht}	Number of block hours short for position h in the bid period t.
E_{ht}	Number of excess block hours for position h in the bid period t.
R_i	Number of months in pay protection paid to pilot $i \in \lambda$.
QA_i	Advance month of the junior pilot who advances the earliest before pilot $i \in \lambda$.
M_i	Min (RF_i,R_i), the total number of months the pilot is pay-protected for the position he holds.
R_{58i}	Number of months in pay protection paid to pilot $i \in 58Y$ (for all pilots 58 years old with bid and no bid).

$$(5) \quad A_{1i} = \begin{cases} 1 & \text{if the minimum}(\mathrm{RF_i}, \mathrm{R_i}) \text{ is } \mathrm{RF_i} \\ 0 & \text{otherwise} \end{cases}$$

$$(6) \quad A_{2i} = \begin{cases} 1 & \text{if the minimum}(\mathrm{RF_i}, \mathrm{R_i}) \text{ is } \mathrm{R_i} \\ 0 & \text{otherwise} \end{cases}$$

MIP Model

Objective Function

$$\text{Minimize} \quad PNH\sum_t\sum_h NHCost_{ht}y_{NHht} + PNA\sum_t\sum_{i\in NA} NACost_{it}y_{it} +$$

$$(7) \quad PF\sum_t\sum_{i\in F} FCost_{it}y_{it} + Ppay(\sum_{i\in\lambda_1} a_iR_i + \sum_{i\in 58Y} a_iR_{58i} + \sum_{i\in\lambda_2} a_iM_i) +$$

$$PS * PBH\sum_h\sum_t S_{ht}/Block_{ht} + PE * PBH * (1/3)\sum_h\sum_t E_{ht}/Block_{ht}$$

Model Constraints

The following constraints along with the objective function of equation (7) make up the core model. Other constraints may be added, as explained in the following section, to account for various user options.

The constraint that all pilots will be trained is represented by the following:

$$(8) \quad \sum_{t\in\phi(i)} y_{it} = 1 \qquad \forall i \in \lambda$$

, where $\forall$ is a symbol which means "for all", $\in$ is a symbol which means "in the set", and thus $\forall i \in \lambda$ means "for all pilots i in the set λ". Therefore, this constraint appears once for each pilot in the set of pilots λ.

To ensure that all pilots who are straight advances to the same position are advanced in seniority order and that all pilots who are straight displacements are advanced in reverse seniority order:

$$(9a) \quad \sum_t ty_{it} - \sum_t ty_{i+1t} \leq 0 \qquad \forall h, i \in SA(h)$$

$$(9b) \quad \sum_t ty_{i+1t} - \sum_t ty_{it} \leq 0 \qquad \forall h, i \in SD(h)$$

In order to ensure that all new hires are hired, the following constraint may be added:

$$(10)\quad \sum_t y_{NHht} = Total_h \qquad \forall h.$$

To ensure that any new hires advance after any existing pilot who is a straight advance for the same position :

$$(11)\quad y_{NHht} - MNH_{ht}\sum_{j=1}^{t} y_{ij} \le 0 \qquad \forall h,t,i \in SA(h)$$

To ensure that all 60 year old no award pilots are released,

$$(12)\quad \sum_t y_{NAit} = 1 \qquad \forall i \in NA.$$

To ensure that all Furlough pilots are released,

$$(13)\quad \sum_t y_{Fit} = 1 \qquad \forall i \in F.$$

To ensure that the capacity of training facilities is not exceeded,

$$(14)\quad \sum_{i\in\varphi(t,f)} y_{i,t+L(i)} \le CAP_{ft} \qquad \forall t,f.$$

In order to compute the shortages in block hours per position,

$$(15)\quad \Big(\sum_{i\in H(h),k\le t} y_{ik} * uti_{hk} - \sum_{i\in H'(h),k\le t} y_{ik} * uti_{hk}\Big) - S_{ht} \le Initial_{ht} - Blockhrs_{ht}(1-\alpha_{ht}) \quad \forall(h,t).$$

To compute the excesses in block hours per position:

$$(16)\quad \Big(\sum_{i\in H(h),k\le t} y_{ik} * uti_{hk} - \sum_{i\in H'(h),k\le t} y_{ik} * uti_{hk}\Big) + E_{ht} \ge Initial_{ht} - Blockhrs_{ht}(1+\beta_{ht}) \quad \forall(h,t).$$

To compute the pay protection for pilots in the training set based on the order in which pilots are trained and advanced, the following three constraints are added:

$$(17a)\quad R_i \geq \Big(\sum_{t\in\phi(i)} ty_{it} - QA_i\Big) \qquad \forall i \in \lambda.$$

$$(18a)\quad QA_i \leq QA_j \qquad \forall i \in \lambda, j->next\_in\_P(i).$$

$$(19a)\quad QA_i \leq \sum_{k\in\phi(j)} ky_{jk} \qquad \forall i \in \lambda, j->next\_in\_P(i).$$

In the case of displaced pilots:

$$(17b)\quad R_i \geq \Big(QA_i - \sum_{t\in\phi(i)} ty_{it}\Big) \qquad \forall i \in \lambda.$$

$$(18b)\quad QA_i \geq QA_j \qquad \forall i \in \lambda, j->next\_in\_P(i).$$

$$(19b)\quad QA_i \geq \sum_{k\in\phi(j)} ky_{jk} \qquad \forall i \in \lambda, j->next\_in\_P(i).$$

To compute the pay protection for pilots in the training set that are already pay protected at the beginning of the planning horizon, or pay protected because they are advanced after the effective date, the following two constraints are added:

If a pilot already is pay-protected then,

$$(20)\quad R_i \geq \Big(\sum_{t\in\phi(i)} ty_{it}\Big) - G_i \qquad \forall i \in APP.$$

Otherwise,

$$(21)\quad R_i \geq \Big(\sum_{t\in\phi(i)} ty_{it} - BidEff_i\Big) \qquad \forall i \in \lambda.$$

To compute the pay protection for pilots 58 years old that could have held a position but the position was not awarded,

$$(22)\quad R_{58i} \geq (N - \sum_{t\in\phi(i)} ty_{jt}) \qquad \forall j \in PP_{58}(i), i \in 58Y.$$

To compute the pay protection for pilots 58 years old that hold a position but could have held a better one, the following four constraints may be applied:

$$(23)\quad RF_i \geq (N - R_{58i} - \sum_{t\in\phi(i)} ty_{it} + R_i) \qquad \forall i \in \lambda_2.$$

$$(24)\quad M_i \geq (R_i - (N+1)A_{1i}) \qquad \forall i \in \lambda_2.$$

$$(25)\quad M_i \geq (RF_i - (N+1)A_{2i}) \qquad \forall i \in \lambda_2.$$

$$(26)\quad A_{1i} + A_{2i} = 1 \qquad \forall i \in \lambda_2.$$

To ensure that pilots 60 years old with no award are released in reverse seniority order,

$$(27)\quad \sum_{t=k}^{N} y_{NAit} - \sum_{t=k}^{N} y_{NAi-1t} \geq 0 \qquad \forall i \in NA, k \in \{1..N\}.$$

To ensure that furlough pilots are released in reverse seniority order,

$$(28)\quad \sum_{t=k}^{N} y_{Fit} - \sum_{t=k}^{N} y_{Fi-1t} \geq 0 \qquad \forall i \in F, k \in \{1..N\}.$$

The following four constraints ensure nonnegative variable values,

$$(29)\quad R_i \geq 0 \qquad \forall i \in \lambda,$$

$$(30)\quad R_{58_i} \geq 0 \qquad \forall i \in 58Y.$$

$$(31)\quad M_t \geq 0 \qquad \forall i \in \lambda_2.$$

$$(32)\quad RF_i \geq 0 \qquad \forall i \in \lambda_2.$$

Modifications To Model Based on Optimization Options

The user may define whether the cancellation of vacations will be allowed. The default position is that no vacations will be cancelled. To consider the vacations, the MIP Model will not include variables y_{it} for months in which pilot i cannot be advanced due to a vacation conflict. If the user wants to cancel vacations, variables will be defined for all months in the MIP Model.

The user may also define whether a particular sub-base will be closed, and the last month pilots can reside in that particular location. To consider the closing of a sub-base, the MIP Model will not define variables y_{it} for the months in which pilot i cannot be advanced due to a conflict with the sub-base closing date.

The user further may define whether a particular sub-base will be opened, and the first month that pilots can be assigned to that particular sub-base. To consider the opening of a sub-base, the MIP Model will not define variables y_{it} for the months in which pilot i cannot be advanced due to a conflict with the sub-base opening date

The user also may define whether a training plan is desired that accommodates the training of sixty year old pilots in the next class after their sixtieth birthday, or in any class during the planning horizon. The default position is to train in the next class after the pilot's sixtieth birthday. To consider the training of the 60 year old pilots, the MIP Model defines variables y_{it} only for the months that correspond to the particular option chosen by the user.

The user may define earliest training start dates for all pilot positions. The model will define variables y_{it} only for the months corresponding to possible training starts for pilots in each position.

A latest advance date for pilots with a given bid effective date may be specified by the user. The model will define variables y_{it} only for the months corresponding to possible training starts for each position.

The user may determine whether to generate a training plan using block hours by fleet or by sub-base. If the user generates a scenario by sub-base, the block hours by sub-base will be computed from the block hours by fleet. The constraints of equations (15) and (16) will be generated by sub-base, and the variables S_{ht} and E_{ht} will be defined by sub-base instead of by fleet.

The user may enforce a training month or a set of training months for a specific pilot. If the user defines the training month or potential set of training months for a particular pilot, the variables y_{it} for each pilot will be defined only for the appropriate month or months.

The user may allow training capacity to be exceeded for each fleet. If the user allows the training capacity for a particular fleet to be exceeded, the variable z_{ft} will be added to the constraint of equation (14) in the MIP Model as follows:

$$(33) \quad \sum_{i\in\varphi(t,f)} y_{i,t+L(i)} - z_{ft} \le CAP_{ft} \qquad \forall t,f\,.$$

The user may define the minimum number of training starts per fleet per bid period. If the user sets a minimum value different than zero for fleet f and bid period t, then the following constraint will be added to the model where MIN_f_t is the minimum number of training starts for fleet f in bid period t..

$$(34) \quad \sum_{i\in NH} y_{it+L(i)} \ge MIN\_f\_t \qquad \forall t,f$$

The user may define the number of new hires per position, or let the optimizer engine 8 of Figure 1 determine the optimal number of new hires. When the optimizer engine 8 is used, the user must define the maximum number of new hires per sub-base and an earliest date for hiring. The

optimizer engine 8 will use as many new hires as needed up to the numbers defined by the user. If the user wants the optimizer engine 8 of Figure 1 to decide an optimal number of new hires, then the equality "="found in the constraint of equation (10) will be changed to an inequality "<=".

If the user wants a solution in which all new hires are clustered in groups of a given number and size and the user defines the total number of new hires, then the optimizer engine 8 will cluster as many groups of the size desired as possible. If the total number of new hires defined by the user is not a multiple of the desired group size, the solution will include one group of size different from the desired group size. If however the optimizer engine 8 defines the optimal number of new hires, and the user wants a solution with new hires clustered in groups, all groups in the solution will be of the desired size.

If the user wants to cluster the new hires into groups, and the user determines the number of new hires, the constraints of equations (35), (36), and (37) below will be added to the MIP Model. In these constraints MM is the desired multiple in which to hire, K_t is an integer variable representing the number of groups of pilots of size MM that will be advanced in month t, KK_t is an integer variable representing new hire pilots not in groups of size MM in month t, P_t is a binary variable for pilots that will not be hired in a group (when the total number of pilots to hire is not a multiple of MM), and Residual is the total new hires modulus MM.

$$(35) \quad \sum_{i \in NH} y_{NHit+L(i)} - KK_t = MM * K_t \qquad \forall t$$

$$(36) \quad KK_t \leq \text{Residual} * P_t \qquad \forall t$$

$$(37) \quad \sum_t P_t = 1$$

If the user wants to cluster the new hires into groups, and the optimizer engine 8 determines the optimal number of new hires, the constraints of equations (38), (39), and (40) below will be added to the model.

$$(38) \quad \sum_{i \in NH} y_{NHit+L(i)} - KK_t = MM * K_t \qquad \forall t$$

$$(39)\quad KK_t \le MM * P_t \qquad \forall t$$

$$(40)\quad \sum_t P_t = 1$$

The user may define the minimum number of new hires per bid period, the maximum number of new hires per bid period, and either a fixed number of new hires per position per month or a minimum number of new hires per position per month. If the user sets a maximum number of new hires per bid period, the following set of constraints will be added to the model, where MAX_NUM_t is the maximum number of new hires in bid period t.

$$(41)\quad \sum_h y_{NHht+L(h)} <= MAX\_NUM\_t \qquad \forall t$$

If the user constrains the minimum number of new hires per bid period, the constraints of equations (42) and (43) will be added to the model, where MIN_NUM_t is the minimum number of new hires in bid period t if any pilot will be hired, MNH_t is a binary variable representing whether any pilot will be hired in bid period t, and $BigM_t$ is the total number of new hires.

$$(42)\quad \sum_h y_{NHht+L(h)} \ge MIN\_NUM\_t * MNH_t \qquad \forall t$$

$$(43)\quad \sum_h y_{NHht+L(h)} \le BigM_t * MNH_t \qquad \forall t$$

The user may minimize percentage shortages and excesses of block hours, or total shortages and excesses in block hours. If the user wants to minimize total shortages and total excess instead of percentage shortage and percentage excess, the objective function will read as follows:

$$\begin{aligned} &Minimize\ PNH\sum_t\sum_h NHCost_{ht}y_{NHht} + PNA\sum_t\sum_{i\in NA} NACost_{it}y_{it} + \\ (44)\quad &PF\sum_t\sum_{i\in F} FCost_{it}y_{it} + Ppay(\sum_{i\in\lambda_1} a_iR_i + \sum_{i\in 58Y} a_iR_{58i} + \sum_{i\in\lambda_2} a_iM_i) + \\ &PS * PBH\sum_h\sum_t S_{ht} + PE * PBH\sum_h\sum_t E_{ht} \end{aligned}$$

The user may restrict the optimizer engine 8 of Figure 1 to build a solution with an absolute limit on the amount of shortage in a given position and a given bid period. If the user defines a maximum total shortage and/or a maximum total percentage shortage for a selected position and bid period, an upper bound will be added to the corresponding S_{ht} variable of the constraint of equation (15). If the user adds both a maximum percentage and a maximum value, the upper bound on the variable will be given by the more restrictive of the two values.

The user may define to what extent the optimizer engine 8 of Figure 1 should enforce class balance constraints by specifying a minimum percentage of ideal captain and first officer groupings for pilots in training classes. The ideal balance of pilots in classes is to have the same number of captains as first officers. This preference for building ideal groupings of pilots in training classes cannot always be enforced due to imbalances in the load of captains and first officers to train, but it can be enforced to the degree possible with the following additional constraints for all fleets. In the following constraints, N_c is the number of captains that need to be trained, N_f is the number of first officers that need to be trained, C_t is an integer variable to track the number of captain imbalances in month t, F_t is an integer variable to track the number of first officers imbalances in month t, %balance is the percent of possible ideal captain and first officer groupings chosen by the user to be enforced in the solution.

If Max $\{N_c, N_f\} = N_c$, then add the following constraint for each fleet and each bid period to track the groupings:

(45) Σ captains going to training in month $t + C_t$
>= Σ first officers going to training in month $t + F_t$

, where the symbol ">=" means greater than or equal to.

Then add the following constraint to enforce the minimum percentage:

(46) $\sum_t C_t \leq N_f(1 - \%balance)$

If Max $\{N_c, N_f\} = N_f$, then add the following constraint for each fleet and each bid period to track the groupings:

(47) Σ captains going to training in month t + C_t

<= Σ first officers going to training in month t + F_t

,where the symbol "<=" means less than or equal to.

Then add the following constraint to enforce the minimum percentage:

$$(48) \quad \sum_t F_t \leq N_c(1 - \%balance)$$

The user may designate whether to train as many pilots as possible in the planning horizon, or to engage the optimizer engine 8 of Figure 1 to determine how many pilots to train before the end of the planning horizon. If the user designates that the engine train as many pilots as possible, an overflow month is used to handle cases where the demand for pilots to train exceeds the capacity to train. When the user defines the percentage of pilots to be trained by the effective date to be less than 100%, the capacity of the overflow month is increased (based upon the percentage given) to allow the option not to train this percentage of the pilots.

The user defines the level of importance of each of the following factors in the MIP Model solution: block hour excess, block hour shortage, pay protection cost, new hires cost, no award cost, and furlough cost represented by the parameters PE, PS, Ppay, PNH, PNA, and PF respectively. When the user modifies the level of importance of any factor, the modification will be reflected in the objective function (7) of the MIP Model by altering the value of the parameters PE, PS, Ppay, PNH, PNA, and PF. By examining the objective function (7), one observes that these parameters act as weights increasing or decreasing the relative importance of their corresponding cost.

The user may further exclude certain costs in the computation of a solution. Costs excluded by the user will be removed from the objective function of the MIP Model as well as appropriate constraints. For example, if the user does not include pay protection cost, then the constraints represented by equations (17b), (18b), (19b), (17b), (18b), (19b), (20), (21), (22), (23), (24), (25) and (26) above will not be included in the MIP Model, and the term

$Ppay(\sum_{i\in\lambda_1} a_i R_i + \sum_{i\in 58Y} a_i R_{58i} + \sum_{i\in\lambda_2} a_i M_i)$ will not be included in the objective function (7) of the MIP Model.

The preprocessing stage of logic step 22 of Figure 2 is illustrated in more detail in Figure 3, where the previously described content of the input data module at logic step 21 is supplied in the form of an XML document to an XML processor. The data supplied consists of general information regarding each pilot in the airline, and specifically each pilot's bid award for the system bid for which a training plan is being created. Figure 3 shows how each pilot's information is sorted in preparation for the model creation stage of logic step 23 of Figure 2.

Referring to Figure 3, the logic flow process enters the optimization engine 8 at logic step 30, and then proceeds to logic step 31 where data from the optimizer data provider 7 of Figure 1 is received. The data includes pilot data such as system bid award information, average pay hours, new hire information, pilot utilization, pay protection, pilot vacation and absence information, and retirement criteria . Other information is also included such as block hour requirements per position each bid period, as well as user options such as the level of importance of each cost factor, and sub-base settings.. With this data, the logic flow process continues to logic step 32 and enters a loop that will analyze the data for each pilot. The other pieces of data do not require processing and are stored for use in the model creation stage of logic step 23 of Figure 2. Through this analysis of pilot data, lists and internal data structures are prepared for the model creation stage of logic step 23.

The logic flow process next advances from logic step 32 to logic step 33 of Figure 3 where the pilot lists are separated into specific types including advancement, retirement, training, no awards, and furloughs. Logic steps 34, 36, 39, 43, and 49 represent logic decisions that separate the pilots by type for appropriate processing. Upon the logic flow process moving from logic step 33 to logic step 34, a determina6on is made as to whether a pilot under consideration is a retirement pilot. If so, the logic flow process proceeds to logic step 35 where pilots who are retiring are placed in a retirement list. The logic flow process then returns to logic step 32 to determine whether there are any further pilots to process, and thereafter continues as before described.

If a retirement pilot is not identified at logic step 34, the logic flow process advances to logic step 36 where no award and furlough pilots are identified. If a no award or furlough pilot is being processed, the logic flow process jumps to logic step 37 where the pilot is placed on a no award or

furlough list, and then continues to logic step 38 to specify possible release months for the pilot. A pilot may be released in any month unless the user has entered a restriction that pilots may only be released in certain months. From logic step 38, the logic flow process loops back to logic step 32 to continue as before described.

If at logic step 36 it is determined that the pilot under consideration is neither a no award nor a furlough pilot, the logic flow process advances to logic step 39 where age 58 pilots are identified. If an age 58 pilot is identified, the logic flow process proceeds to logic step 40 where the pilot is placed on an age 58 pilot list, and then continues to logic step 41 to determine which pilots may cause pay protection for the pilot under consideration. The logic flow process then moves to logic step 42 where it is determined whether the pilot under consideration requires advancement or training. If not, the logic flow process loops back to logic step 32 to continue as before described. However, if the age 58 pilot does require advancement or training, the flow process proceeds to logic step 44 as more particularly described below.

If at logic step 39 no 58 year old pilot is identified, the logic flow process advances to logic step 43 where advancement and training pilots are identified. If an advancement or training pilot is identified, the logic flow process moves to logic step 44 where the pilot is placed on an advancement or training list, and then continues to logic step 45 where possible advancement months are determined. Pilot advancement months are limited by their awarded vacation, user preferences, sub-base openings and closings, and available training resources. From logic step 45, the logic flow process advances to logic step 46 where it is determined whether there are any possible assignments for the pilot under consideration. If there are none, the logic flow process jumps to logic step 50 where the pilot is placed on an exception list. The exception list includes pilots who have no advancement months and thus no possible assignment, or who have inconsistent data. Following logic step 50, the logic flow process loops back to logic step 32 to continue as before described. If it is determined at logic step 46 that there are possible assignments for the pilot under consideration, the logic flow process proceeds from logic step 46 to logic step 47 where it is determined whether the pilot requires training. If the pilot does require training, the logic flow process moves from logic step 47 to logic step 48 where a list of possible classes the pilot can attend is generated, and then loops back to logic step 32 to continue as before described. If it is determined at logic step 47 that the pilot does not require training, the logic flow process loops back to logic step 32 to continue as before described.

If at logic step 43 it is determined that the pilot under consideration does not require training or advancement, the logic flow process advances to logic step 49 where pilots who have a new bid award but have not been separated are identified. If the pilot under consideration has a new bid award, the logic flow process advances to logic step 50 to continue as before described. If, however, it is determined at logic step 49 that the pilot has no new bid award, the logic flow process loops back to logic step 32 to continue as before described. When it is determined at logic step 32 that all pilots have been processed, the logic flow process proceeds to logic step 51 where new hire options chosen by the user are processed and a list of possible new hire months is generated. From logic step 51 the logic flow process continues to logic step 52 to transfer to the model creation stage 23 of Figure 2 and perform in accordance with the logic flow diagram of Figure 4

Referring to Figure 4, the logic flow begins at logic step 60 and continues to logic step 61 where the preprocessed data prepared in the preprocessing stage of Figure 3 is used to construct all of the variables and constraints needed to build the MIP Model (as described above) in view of the options chosen by the user. The left most column of logic steps is directed to the creation of variables, and the right most column of logic steps is directed to the creation of constraints.

Each variable that is created is a combination of a key and a value, and has a numeric type, a class type, and a list of the constraints of which it is a member. A key uniquely identifies a variable. For example, the key for a variable describing the advancement of a pilot is a combination of the pilot's ID and the month in which the pilot can advance. For variables describing the shortages in block hours for training, the triple of fleet, status and month is the key. The value of a variable refers to the value given upon solution of the MIP Model. The numeric type refers to the binary, integer, and continuous variable types. The class type of a variable refers to types such as assignment, block hour variable, and class balance variable.

Each constraint that is created is logically grouped into a constraint set, where each set corresponds to a specific class of constraints. For example, all constraints in the system that enforce the requirement that pilots have to be released in reverse seniority order can belong to a single constraint set. Whenever a constraint is added to a constraint set, all the variables that are part of the constraint are identified, and the constraint list for each variable is modified to include the constraint.

From logic step 61 of Figure 4, the logic flow process continues to logic step 62 where assignment variables are created from the lists previously created in the preprocessing stage of Figure 3 for pilots who are training and advancing, no awards, furloughs, and new hires. The logic flow process then proceeds to logic step 63 to determine whether training capacities can be exceeded. If training capacity can be exceeded, the logic flow process continues to logic step 64 to create excess capacity variables for use in the capacity constraint of equation (33). If it is determined at logic step 63 that training capacity will not be exceeded, the logic flow process proceeds to logic step 65 where variables are created to track block hour shortages and excesses. The logic flow process then continues to logic step 66 where it is determined whether pay protection is possible. If so, the logic flow process advances to logic step 67 to create all pay protection variables. After logic step 67, or if it is determined at logic step 66 that no pay protection is possible, the logic flow process moves to logic step 68 where it is determined whether there are any possible new hires. If so, the logic flow process continues to logic step 69. If not, the logic flow process moves to logic step 70.

At logic step 69, if new hires are to be clustered into groups of a given size as specified by the user, the logic flow process continues to logic step 71 where new hire clustering variables are created. After logic step 71, or if no new hire clustering into groups has been specified at logic step 69, the logic flow process proceeds to logic step 72 where it is determined whether new hire minimums will be enforced. If yes, the logic flow process proceeds to logic step 73 where minimum new hire variables are created. From logic step 73, or if new hire minimums are not enforced at logic step 72, the logic flow process moves to logic step 70 where class balance variables are created.

From logic step 70, the logic flow process continues to logic step 74 where the creation of constraints begins with the assignment constraints of equations (8), (9a), (9b), (10), (11), (12), and (13). These constraints ensure that each pilot will be assigned a time to train, advance, be released, or be hired by the MIP Model. Following logic step 74, the logic flow process continues to logic step 75 where the capacity constraint of equation (14) or (33) is created, based upon the users specification, to allow the model to exceed or enforce training capacities. The logic flow process then continues to logic step 76 to create the constraints of equations (15) and (16) that track shortages and excesses in block hours at each position and each bid period respectively. The logic flow process thereafter moves to logic step 77 to determine whether pay protection is possible. If pay protection is possible, the logic flow process continues to logic step 78 to create the constraints of equations (17a), (18a), (19a), (17b), (18b), (19b), (20), (21), (22), (23), (24), (25), and (26), to

manage pay protection for pilots who are advancing, displaced, age 58, and already pay protected. After logic step 78, or it is determined at logic step 77 that no pay protection is possible, the logic flow process moves to logic step 79 where the creation of the constraints of equations (25) and (26) takes place. These constraints manage the release of no award and furlough pilots.

From logic step79, the logic flow process proceeds to logic step 80 where the minimum capacity constraints of equation (34) are created. Thereafter, the logic flow process moves to logic step 81 where if no possible new hires are specified, the logic flow process continues to logic step 82. If new hires are specified at logic step 81, however, the logic flow process continues to logic step 83. If new hire clustering into groups is specified at logic step 83, the logic flow process proceeds to logic step 84. At logic step 84, the new hire clustering constraints of equations (35), (36), and (37) are created if the user defines the exact number of new hires. If the user specifies that the optimizer engine 8 of Figure 1 will define the optimal number of new hires, the constraints of equations (38), (39), and (40) are created to manage the clustering of new hires. From logic step 84, or if no clustering of new hires is specified at logic step 83, the logic flow process moves to logic step 85 where it is determined whether new hire maximums are to be enforced. If yes, the logic flow process advances to logic step 86 to create the maximum new hire constraints of equation (41), and then continues to logic step 87. If it is determined at logic step 85 that the new hire maximum is not to be enforced, the logic flow process jumps to logic step 87 to determine whether the minimum new hire constraints are to be enforced. If yes, the logic flow process proceeds to logic step 88 where the constraints of equations (42) and (43) that bound the minimum number of new hires are generated. From logic step 88, or if it is determined at logic step 87 that minimum new hire constraints will not be enforced, the logic flow process moves to logic step 82 where the class balance constraints of equations (45) and (46) are created if there are more captains training than first officers, and the constraints of equations (47) and (48) are created if there are more first officers than captains training. After logic step 82, the logic flow process moves to logic step 89 to transfer to the solution and postprocessing stage of logic step 24 of Figure 2 and perform in accordance with the logic flow diagram of Figure 5.

Referring to Figure 5, the logic flow process begins at logic step 100, and proceeds to logic step 101 with the MIP Model prepared in the model creation stage of logic step 23 of Figure 2. The MIP Model is solved and the results are used to derive a training plan to display to the user. The

leftmost column of logic steps in Figure 5 describes the model solution process, and the rightmost column of logic steps illustrate the derivation of a training plan from the model solution.

From logic step 101, the logic flow process proceeds to logic step 102 where the linear programming relaxation of the MIP Model (LP Model) is solved. The LP Model is obtained by relaxing the integrality of all binary and integer variables in the MIP Model, i.e., by treating them as continuous variables instead of binary and integer variables. This relaxation of variables results in a mathematical problem that can be solved much more efficiently than the original MIP Model. Our experience indicates that the resulting LP Model can be solved in seconds using commercially available software development kits and runtime libraries such as ILOG Concert Technology 1.0 and ILOG CPLEX 7.0 sold by ILOG, Inc. of Paris, France. The invention integrates such commercially available software development kits and runtime libraries to solve the mathematical models in logic step 102 and 106.

From logic step 102, the logic flow process continues to logic step 103 to determine whether a feasible solution exists for the LP Model solved in logic step 102. If a feasible solution exists at logic step 103, the logic flow process advances to logic step 104. If a feasible solution does not exist at logic step 103, however, the logic flow process jumps to logic step 105 where output informing the user of the infeasible solution is generated.

An infeasible problem occurs when constraints are defined in such a way that there is no way to assign a value to every variable and still satisfy all of the constraints. A simple example is illustrated with the following two constraints bounding a continuous variable x: $x < 5$ and $x > 7$. There is no assignment for the variable x that will satisfy both constraints. For the LP Model solved at logic step 102 some possible reasons for infeasibility are listed below: (1) the user specifies a bound on shortages for a specific position and bid period, but there are not enough pilots assigned to that position to meet the bound; (2) the training capacity is not sufficient to train all pilots by the effective date chosen; and (3) there is sufficient capacity to train a group of pilots spread out over a number of months, but the pilots have very limited months in which to train due to vacation, or closing and opening of sub-bases.

At logic step 104, the LP Model solution of logic step 102 is used to modify the block hour shortage and excess costs, PS and PE respectively, used in the objective function of the MIP Model.

The LP Model provides an estimate of the dollar costs of the MIP Model and of the block hour shortages and excesses. The dollar costs in the model are the costs relating to pay protection, new hires, no awards, and furloughs, Ppay, PNH, PNA, and PF respectively. To place higher emphasis on meeting block hours than on the dollar costs in the MIP Model, the block hour costs are modified to have a value roughly twice as large as the dollar costs based on the solution to the LP Model. From logic step 104, the logic flow process continues to logic step 106 where the MIP Model is solved. Our experience indicates that the MIP Model can be solved in less than an hour using commercially available software development kits and runtime libraries as described above.

The logic flow process next proceeds from logic step 106 to logic step 107 where a determination is made as to whether a feasible solution was found. If yes, the logic flow process jumps to logic step 108 to commence the development of a training plan from the solution to the MIP Model.

At logic step 108, the assigned variable values from the MIP Model solution are transformed into lists showing pilot training, advancement, no awards and furlough releases, new hires, and pay protected pilots. After logic step 108, the logic flow process advances to logic step 109 where headcounts are updated based upon the timing of training, advancement, no awards and furlough releases, and new hires to show how many pilots will be in each position over time. The logic flow process then continues to logic step 110 where block hour capability is calculated by multiplying the updated headcount by pilot utilization values for each position and bid period. The block hour capability shows how much work can be done in each position in each bid period. This capability is then compared to the required block hours for each position in each bid period to determine any block hour shortages or excesses. Following logic step 110, the logic flow process advances to logic step 111 where all pilots who are going to training are placed on class rosters in the bid period determined by the model. Thereafter, the logic flow process continues to logic step 112 where the dollar costs associated with the solution are computed as shown below in equations (49) - (53).

(49) No Award Cost = $PNA\sum_{t}\sum_{i\in NA} NACost_{it} y_{it}$

(50) Furlough Cost = $PF\sum_{t}\sum_{i\in F} FCost_{it} y_{it}$

(51) New Hire Cost = $PNH\sum_{t}\sum_{i\in NH} NHCost_{it} y_{it}$

(52) Pay Protection Cost= $Ppay(\sum_{i\in\lambda_1} a_i R_i + \sum_{i\in 58Y} a_i R_{58i} + \sum_{i\in\lambda_2} a_i M_i)$

(53) Total Cost = No Award cost + Furlough cost + New Hire Cost + Pay Protection Cost.

The logic flow process then proceeds from logic step 112 to logic step 105 where the solution information generated in logic steps 108-112 is transformed into training plan reports for access by the user.

From logic step 105, the logic flow process moves to logic step 113 to end the process.

The present invention has been particularly shown and described in detail with reference to a preferred embodiment, which is merely illustrative of the principles of the invention and is not to be taken as a limitation to its scope. It further will be readily understood by those skilled in the art, operations research, that substitution of equivalent elements, reordering of steps, and other modifications and alterations of the invention may occur without departing from the scope and spirit of the invention. The appended claims are intended to include within their scope such modifications and alterations.

<u>WHAT IS CLAIMED IS:</u>